



`^\\d+.*`

# JavaScript Regular Expressions

SENG 4640

Software Engineering for Web Apps

Winter 2023

Sina Keshvadi

Thompson Rivers University

# Review

---

- JavaScript strings are sequences of characters
- JavaScript strings are immutable
- Strings are objects and have their own functions

# Strings and Characters

---

- We can get the number of characters in a string using the `length` property
- We can access each character by its (0-based) index using `charAt` or array notation

# Strings and Characters

---

- We can get the number of characters in a string using the `length` property
- We can access each character by its (0-based) index using `charAt` or array notation

```
var name = 'toucan';
```

# Strings and Characters

---

- We can get the number of characters in a string using the **length** property
- We can access each character by its (0-based) index using **charAt** or array notation

```
var name = 'toucan';  
  
name.length; // 6
```

# Strings and Characters

---

- We can get the number of characters in a string using the **length** property
- We can access each character by its (0-based) index using **charAt** or array notation

```
var name = 'toucan';  
  
name.length; // 6  
  
name.charAt(3); // 'c'
```

# Strings and Characters

---

- We can get the number of characters in a string using the **length** property
- We can access each character by its (0-based) index using **charAt** or array notation

```
var name = 'toucan';  
  
name.length; // 6  
  
name.charAt(3); // 'c'
```

# Strings and Characters

---

- We can get the number of characters in a string using the **length** property
- We can access each character by its (0-based) index using **charAt** or array notation

```
var name = 'toucan';  
  
name.length; // 6  
  
name.charAt(3); // 'c'  
name[3]; // 'c'
```



# Strings and Characters

---

- We can get the number of characters in a string using the **length** property
- We can access each character by its (0-based) index using **charAt** or array notation

```
var name = 'toucan';  
  
name.length; // 6  
  
name.charAt(3); // 'c'  
name[3]; // 'c'
```

- Remember! JavaScript strings are immutable!

# Strings and Characters

---

- We can get the number of characters in a string using the **length** property
- We can access each character by its (0-based) index using **charAt** or array notation

```
var name = 'toucan';  
  
name.length; // 6  
  
name.charAt(3); // 'c'  
name[3]; // 'c'
```

- Remember! JavaScript strings are immutable!

```
var animal = 'cat';
```

# Strings and Characters

---

- We can get the number of characters in a string using the **length** property
- We can access each character by its (0-based) index using **charAt** or array notation

```
var name = 'toucan';  
  
name.length; // 6  
  
name.charAt(3); // 'c'  
name[3]; // 'c'
```

- Remember! JavaScript strings are immutable!

```
var animal = 'cat';  
  
animal[0] = 'r';
```

# Strings and Characters

---

- We can get the number of characters in a string using the **length** property
- We can access each character by its (0-based) index using **charAt** or array notation

```
var name = 'toucan';  
  
name.length; // 6  
  
name.charAt(3); // 'c'  
name[3]; // 'c'
```

- Remember! JavaScript strings are immutable!

```
var animal = 'cat';  
  
animal[0] = 'r';  
  
console.log(animal); // still 'cat'
```

# Modifying Strings

---

- We can modify a string but these functions return a **new** string (since strings are immutable!)

# Modifying Strings

---

- We can modify a string but these functions return a **new** string (since strings are immutable!)

```
var friend = 'turtle';
```

# Modifying Strings

---

- We can modify a string but these functions return a **new** string (since strings are immutable!)

```
var friend = 'turtle';  
  
friend.toUpperCase(); // 'TURTLE'
```

# Modifying Strings

---

- We can modify a string but these functions return a **new** string (since strings are immutable!)

```
var friend = 'turtle';  
  
friend.toUpperCase(); // 'TURTLE'  
console.log(friend); // 'turtle'
```



# Modifying Strings

---

- We can modify a string but these functions return a **new** string (since strings are immutable!)

```
var friend = 'turtle';  
  
friend.toUpperCase();    // 'TURTLE'  
console.log(friend);    // 'turtle'
```

```
var message = '_hello everyone_' ;
```

# Modifying Strings

---

- We can modify a string but these functions return a **new** string (since strings are immutable!)

```
var friend = 'turtle';  
  
friend.toUpperCase(); // 'TURTLE'  
console.log(friend); // 'turtle'
```

```
var message = ' hello everyone '  
message = message.trim(); // 'hello everyone'
```

# Modifying Strings

---

- We can modify a string but these functions return a **new** string (since strings are immutable!)

```
var friend = 'turtle';  
  
friend.toUpperCase(); // 'TURTLE'  
console.log(friend); // 'turtle'
```

```
var message = ' hello everyone ';  
message = message.trim(); // 'hello everyone'
```

```
var myAnimal = 'cat'.concat('mouse');
```

# Modifying Strings

---

- We can modify a string but these functions return a **new** string (since strings are immutable!)

```
var friend = 'turtle';  
  
friend.toUpperCase(); // 'TURTLE'  
console.log(friend); // 'turtle'
```

```
var message = ' hello everyone ';  
message = message.trim(); // 'hello everyone'
```

```
var myAnimal = 'cat'.concat('mouse');  
console.log(myAnimal); // 'catmouse'
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith(`programming`);    // true
msg.startsWith(`PROGRAMMING`);    // false

msg.endsWith(`is fun`);            // true

msg.includes(`JavaScript`);        // true
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = 'programming in JavaScript is  
fun';  
msg.startsWith('programming'); // true  
msg.startsWith('PROGRAMMING'); // false  
  
msg.endsWith('is fun'); // true  
  
msg.includes('JavaScript'); // true
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith('programming'); // true
msg.startsWith('PROGRAMMING'); // false

msg.endsWith('is fun'); // true

msg.includes('JavaScript'); // true
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = 'programming in JavaScript is  
fun';  
msg.startsWith('programming'); // true  
msg.startsWith('PROGRAMMING'); // false  
  
msg.endsWith('is fun'); // true  
  
msg.includes('JavaScript'); // true
```



# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith('programming'); // true
msg.startsWith('PROGRAMMING'); // false
msg.endsWith('is fun'); // true
msg.includes('JavaScript'); // true
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = 'programming in JavaScript is fun';  
msg.startsWith('programming'); // true  
msg.startsWith('PROGRAMMING'); // false  
msg.endsWith('is fun'); // true  
msg.includes('JavaScript'); // true
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is  
fun`;  
msg.startsWith(`programming`); // true  
msg.startsWith(`PROGRAMMING`); // false  
  
msg.endsWith(`is fun`); // true  
  
msg.includes(`JavaScript`); // true
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith(`programming`);    // true
msg.startsWith(`PROGRAMMING`);    // false

msg.endsWith(`is fun`);            // true

msg.includes(`JavaScript`);        // true
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = 'programming in JavaScript is
fun';
msg.startsWith('programming'); // true
msg.startsWith('PROGRAMMING'); // false

msg.endsWith('is fun'); // true

msg.includes('JavaScript'); // true
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith(`programming`);    // true
msg.startsWith(`PROGRAMMING`);    // false

msg.endsWith(`is fun`);           // true

msg.includes(`JavaScript`);       // true
```

- We can also get the starting index of a contained substring

```
var title = `the title of my book`;
var start = title.search(`title`); // 4
start = title.search(`banana`);   // -1
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = 'programming in JavaScript is fun';  
msg.startsWith('programming'); // true  
msg.startsWith('PROGRAMMING'); // false  
  
msg.endsWith('is fun'); // true  
  
msg.includes('JavaScript'); // true
```

- We can also get the starting index of a contained substring

```
var title = 'the title of my book';  
var start = title.search('title'); // 4  
start = title.search('banana'); // -1
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith(`programming`);    // true
msg.startsWith(`PROGRAMMING`);    // false

msg.endsWith(`is fun`);           // true

msg.includes(`JavaScript`);       // true
```

- We can also get the starting index of a contained substring

```
var title = `the title of my book`;
var start = title.search(`title`); // 4
start = title.search(`banana`);   // -1
```



# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith(`programming`);    // true
msg.startsWith(`PROGRAMMING`);    // false

msg.endsWith(`is fun`);            // true

msg.includes(`JavaScript`);        // true
```

- We can also get the starting index of a contained substring

```
var title = `the title of my book`;
var start = title.search(`title`); // 4
start = title.search(`banana`); // -1
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith(`programming`);    // true
msg.startsWith(`PROGRAMMING`);    // false

msg.endsWith(`is fun`);           // true

msg.includes(`JavaScript`);       // true
```

- We can also get the starting index of a contained substring

```
var title = `the title of my book`;
var start = title.search(`title`); // 4
start = title.search(`banana`); // -1
```

# Searching Strings

---

- We can determine whether a string starts with, ends with, or includes other strings

```
var msg = `programming in JavaScript is
fun`;
msg.startsWith(`programming`);    // true
msg.startsWith(`PROGRAMMING`);    // false

msg.endsWith(`is fun`);            // true

msg.includes(`JavaScript`);        // true
```

- We can also get the starting index of a contained substring

```
var title = `the title of my book`;
var start = title.search(`title`); // 4
start = title.search(`banana`); // -1
```

# Regular Expressions

---

- A **regular expression** is a pattern of characters
- A string **matches** a regular expression if it adheres to the same pattern
- Example: “consists of exactly three digits (0-9)”
  - ‘123’ matches
  - ‘abc’ does not match
  - ‘12’ does not match
  - ‘12345’ does not match

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/);
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13
```



# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's `search` function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13  
  
status.search(/very/);
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13  
  
status.search(/very/); // -1
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13  
  
status.search(/very/); // -1  
  
status.search(/very/i);
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13  
  
status.search(/very/); // -1  
  
status.search(/very/i); // 13
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13  
  
status.search(/very/); // -1  
  
status.search(/very/i); // 13
```

- Or, we can use the regex's **test** function

```
/script/.test('javascript is so much fun!'); // true
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13  
  
status.search(/very/); // -1  
  
status.search(/very/i); // 13
```

- Or, we can use the regex's **test** function

```
/script/.test('javascript is so much fun!'); // true
```

# Simple Regular Expression Matching

---

- We can pass a regular expression to the string's **search** function to see if it matches the pattern
- In general, it is considered a match if **any** part of the string matches the regular expression

```
var status = 'I am working VERY hard';  
  
status.search(/VERY/); // 13  
  
status.search(/very/); // -1  
  
status.search(/very/i); // 13
```

- Or, we can use the regex's **test** function

```
/script/.test('javascript is so much fun!'); // true
```



# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/ [012] /);
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/ [012] /);
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/ [012] /); // 4
```



# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);           // 4  
/[012]/.test(numbers);
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);           // 4  
/[012]/.test(numbers);           // true
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);           // 4  
/[012]/.test(numbers);             // true
```

- Or ranges of characters or special characters

```
var password = 'password4real';
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);           // 4  
/[012]/.test(numbers);             // true
```

- Or ranges of characters or special characters

```
var password = 'password4real';  
password.search(/[a-z]/);
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);           // 4  
/[012]/.test(numbers);             // true
```

- Or ranges of characters or special characters

```
var password = 'password4real';  
password.search(/[a-z]/);          // 0
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);           // 4  
/[012]/.test(numbers);           // true
```

- Or ranges of characters or special characters

```
var password = 'password4real';  
password.search(/[a-z]/);           // 0  
password.search(/\d/);
```

# Specifying Ranges of Characters

---

- We can also specify multiple valid characters that we want to consider for matching
- For instance, we can look for specific characters

```
var numbers = '5 8 2 5 7 6';  
numbers.search(/[012]/);           // 4  
/[012]/.test(numbers);             // true
```

- Or ranges of characters or special characters

```
var password = 'password4real';  
password.search(/[a-z]/);          // 0  
password.search(/\d/);              // 8
```

# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/[0-9][a-z][0-9]/);
```



# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/[0-9][a-z][0-9]/);
```

# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/[0-9][a-z][0-9]/);
```

# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/ [0-9] [a-z] [0-9] /);
```

# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/ [0-9] [a-z] [0-9] /); // 5
```

# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/[0-9][a-z][0-9]/); // 5
```

- Or look for characters **not** in a range

```
var chars = 'abc123K456';  
chars.search(/[^0-9a-z]/);
```

# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/[0-9][a-z][0-9]/); // 5
```

- Or look for characters **not** in a range

```
var chars = 'abc123K456';  
chars.search(/[^0-9a-z]/);
```

# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/[0-9][a-z][0-9]/); // 5
```

- Or look for characters **not** in a range

```
var chars = 'abc123K456';  
chars.search(/^0-9a-z/);
```

# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/[0-9][a-z][0-9]/); // 5
```

- Or look for characters **not** in a range

```
var chars = 'abc123K456';  
chars.search(/[^0-9a-z]/);
```



# Using Ranges

---

- We can combine different ranges

```
var code = 'abc123d4e5';  
code.search(/[0-9][a-z][0-9]/); // 5
```

- Or look for characters **not** in a range

```
var chars = 'abc123k456';  
chars.search(/[^0-9a-z]/); // 6
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b');
```

```
/[a-z][0-9]?[a-z]/.test('abc');
```

```
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b');
```

```
/[a-z][0-9]?[a-z]/.test('abc');
```

```
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b');
```

```
/[a-z][0-9]?[a-z]/.test('abc');
```

```
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b');
```

```
/[a-z][0-9]?[a-z]/.test('abc');
```

```
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b');
```

```
/[a-z][0-9]?[a-z]/.test('abc');
```

```
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc');  
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true
```

```
/[a-z][0-9]?[a-z]/.test('abc');
```

```
/[a-z][0-9]?[a-z]/.test('a123b');
```



# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc');  
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
  
/[a-z][0-9]?[a-z]/.test('abc');  
  
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc');  
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');
```



# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b'); // false
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');// false
```

- Or optional **multiple** occurrences

```
/[a-z][0-9]*[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');// false
```

- Or optional **multiple** occurrences

```
/[a-z][0-9]*[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');// false
```

- Or optional **multiple** occurrences

```
/[a-z][0-9]*[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');// false
```

- Or optional **multiple** occurrences

```
/[a-z][0-9]*[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');// false
```

- Or optional **multiple** occurrences

```
/[a-z][0-9]*[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');// false
```

- Or optional **multiple** occurrences

```
/[a-z][0-9]*[a-z]/.test('a123b');
```

# Quantifiers

---

- We may want to know whether the string contains an optional **single** occurrence

```
/[a-z][0-9]?[a-z]/.test('a1b'); // true  
/[a-z][0-9]?[a-z]/.test('abc'); // true  
/[a-z][0-9]?[a-z]/.test('a123b');// false
```

- Or optional **multiple** occurrences

```
/[a-z][0-9]*[a-z]/.test('a123b'); // true
```



# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');  
 /^[a-z][0-9]/.test('ab12');  
  
/[a-z][a-z]$/.test('123abc');  
/[a-z][a-z]$/.test('123abc456');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');  
 /^[a-z][0-9]/.test('ab12');  
  
/[a-z][a-z]$/.test('123abc');  
/[a-z][a-z]$/.test('123abc456');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
/^ [a-z] [0-9] /.test ( 'a1b' );  
/^ [a-z] [0-9] /.test ( 'ab12' );  
  
/[a-z] [a-z] $/.test ( '123abc' );  
/[a-z] [a-z] $/.test ( '123abc456' );
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
/^ [a-z] [0-9] /.test('a1b');  
/^ [a-z] [0-9] /.test('ab12');  
  
/[a-z] [a-z] $/.test('123abc');  
/[a-z] [a-z] $/.test('123abc456');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b'); // true
 /^[a-z][0-9]/.test('ab12');

 /[a-z][a-z]$/.test('123abc');
 /[a-z][a-z]$/.test('123abc456');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b'); // true  
 /^[a-z][0-9]/.test('ab12');  
  
/[a-z][a-z]$/.test('123abc');  
/[a-z][a-z]$/.test('123abc456');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false

 /[a-z][a-z]$/.test('123abc');
 /[a-z][a-z]$/.test('123abc456');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false
 /[a-z][a-z]$/.test('123abc');
 /[a-z][a-z]$/.test('123abc456');
```



# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false
 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');     // false
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false
 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false
 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');     // false
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false
 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');     // false
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b'); // true
 /^[a-z][0-9]/.test('ab12'); // false

 /[a-z][a-z]$/.test('123abc'); // true
 /[a-z][a-z]$/.test('123abc456'); // false
```

- This lets us detect **exact** matches

```
 /^[a-z][0-9][a-z]$/.test('a1b');

 /^[a-z][0-9][a-z]$/.test('a1b2c');

 /^[a-z][0-9a-z]*[a-z]$/.test('a1b2c');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false

 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');     // false
```

- This lets us detect **exact** matches

```
 /^[a-z][0-9][a-z]$/.test('a1b');
 /^[a-z][0-9][a-z]$/.test('a1b2c');
 /^[a-z][0-9a-z]*[a-z]$/.test('a1b2c');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
/^[a-z][0-9]/.test('a1b'); // true
/^[a-z][0-9]/.test('ab12'); // false

/[a-z][a-z]$/.test('123abc'); // true
/[a-z][a-z]$/.test('123abc456'); // false
```

- This lets us detect **exact** matches

```
/^[a-z][0-9][a-z]$/.test('a1b'); // true
/^[a-z][0-9][a-z]$/.test('a1b2c');
/^[a-z][0-9a-z]*[a-z]$/.test('a1b2c');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false

 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');     // false
```

- This lets us detect **exact** matches

```
 /^[a-z][0-9][a-z]$/.test('a1b');     // true
  /^[a-z][0-9][a-z]$/.test('a1b2c'); 
 /^[a-z][0-9a-z]*[a-z]$/.test('a1b2c');
```



# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false

 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');     // false
```

- This lets us detect **exact** matches

```
 /^[a-z][0-9][a-z]$/.test('a1b');     // true
 /^[a-z][0-9][a-z]$/.test('a1b2c');   // false
 /^[a-z][0-9a-z]*[a-z]$/.test('a1b2c');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false

 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');     // false
```

- This lets us detect **exact** matches

```
 /^[a-z][0-9][a-z]$/.test('a1b');     // true
 /^[a-z][0-9][a-z]$/.test('a1b2c');   // false

  /^[a-z][0-9a-z]*[a-z]$/.test('a1b2c'); 
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
 /^[a-z][0-9]/.test('a1b');           // true
 /^[a-z][0-9]/.test('ab12');          // false

 /[a-z][a-z]$/.test('123abc');        // true
 /[a-z][a-z]$/.test('123abc456');     // false
```

- This lets us detect **exact** matches

```
 /^[a-z][0-9][a-z]$/.test('a1b');     // true
 /^[a-z][0-9][a-z]$/.test('a1b2c');   // false

 /^[a-z][0-9a-z]*[a-z]$/.test('a1b2c');
```

# startsWith and endsWith Matches

---

- Regular expressions can tell us if a string **contains** a pattern, but we may want to know if the string **starts** or **ends** with the pattern

```
/^[a-z][0-9]/.test('a1b'); // true
/^[a-z][0-9]/.test('ab12'); // false

/[a-z][a-z]$/.test('123abc'); // true
/[a-z][a-z]$/.test('123abc456'); // false
```

- This lets us detect **exact** matches

```
/^[a-z][0-9][a-z]$/.test('a1b'); // true
/^[a-z][0-9][a-z]$/.test('a1b2c'); // false

/^[a-z][0-9a-z]*[a-z]$/.test('a1b2c'); // true
```

# Summary

---

- JavaScript strings are immutable but provide functions that allow us to create new, modified versions of them
- Strings have **startsWith**, **endsWith**, **includes**, and **search** functions
- We can also use regular expressions' **test** function to check for matches in a string

Let's Practice **Regex**